

Fiche technique - API PestAI

Présentation

API FastAPI permettant la détection et classification des maladies des feuilles et ravageurs sur images agricoles, basée sur des modèles YOLO pour la détection et des modèles InceptionV3 et EfficientNet-B0(V3) pour la classification.

Endpoints Principaux

1. POST /detect-image

Description

Analyse une image uploadée pour détecter et classifiez feuilles malades ou ravageurs.

- Paramètres Query
 - - type (string, obligatoire) :
 - - "feuille" pour détection maladie des feuilles
 - - "ravageur" pour détection d'insectes nuisibles
- Paramètres Form-data
 - - file (fichier image, obligatoire) : Image au format jpeg, png, bmp, tiff ou webp.

Réponse (JSON)

```
{
  "filename": "nom_fichier.jpg",
  "width": 1024,
  "height": 768,
  "detections": [
    {
      "bbox": [x1, y1, x2, y2],
      "confidence": 0.87,
      "label": "feuille",
      "crop_url": "http://host/crop/crop_nom_fichier_x1_y1.jpg",
      "classification": {
        "predicted_class": "Tomato__Tomato_Bacterial_spot",
        "confidence": 0.92
      }
    }
  ],
}
```

```
...  
]  
}
```

- Notes
 - - Les bbox sont des coordonnées en pixels dans l'image d'origine.
 - - La classification est facultative et donne la classe prédite sur le crop avec un seuil minimal de confiance.
 - - Si la classification est incertaine, "predicted_class": "incertain" sera retourné.
 - - Crop URL permet de récupérer l'image recadrée.
- Codes HTTP possibles
 - - 200 : succès
 - - 400 : type inconnu, image invalide
 - - 415 : format image non supporté
 - - 500 : erreur interne (modèle non chargé)

2. GET /crop/{filename}

Description

Récupérer une image crop (recadrée) générée par la détection.

- Paramètres Path
 - - filename : nom du fichier crop (ex : crop_nomfichier_100_50.jpg)

Réponse

- - Image JPEG du crop.
- Codes HTTP possibles
 - - 200 : image retournée
 - - 404 : fichier non trouvé

Modèles utilisés

Type	Modèle détection (YOLO ONNX)	Modèle classification (Keras)	Input classification	Classes top (exemples)
Feuille	models/ modele_feuille/	models/ modele_feuille/	384x384 (ou	Tomato_Yellow_ Leaf_Curl_Virus,

	detection/ last.onnx	classification/ model_final.keras	299x299)	Citrus_greening, Tomato_Bacterial_spot, etc.
Ravageur	models/ modele_ravageur/ detection/ best_detection_ravageur.onnx	models/ modele_ravageur/ r/ classification/ best_model.keras	224x224	ants, bees, beetle, catterpillar, moth, slug, wasp, etc.

Paramètres globaux importants

- Seuil détection (confidence): 0.4
- Seuil classification: 0.6 (modifiable par classe)
- Taille minimale des crops pour classification: 50x50 pixels
- Types image supportés : jpeg, png, bmp, tiff, webp
- Suppression automatique des crops après 1 heure (3600 sec)
- Utilisation CPU par défaut ("DEVICE=\"cpu\"", GPU possible si disponible)

Fonctionnalités techniques

- Chargement automatique des modèles YOLO et Keras au démarrage.
- Nettoyage périodique automatique des images crop pour éviter surcharge disque.
- Pré-traitement image adapté à chaque modèle (EfficientNet, InceptionV3, ou autre).
- Détection et classification sur chaque crop détecté.
- API CORS ouverte (origines et méthodes libres).

Limitations / Notes

- API attend des images valides et relativement propres pour de bons résultats.
- Classification peut retourner "incertain" si confiance insuffisante.
- Temps de réponse dépend de la taille image et du nombre d'objets détectés.
- Support GPU possible mais à configurer manuellement.
- Crop URLs générés à partir de l'URL de base, à adapter selon déploiement.

Contact / Support

- Version API : 3.0
- Framework : FastAPI
- Framework ML : YOLO (Ultralytics), TensorFlow Keras
- Auteur : Ibrahima Gabar Diop